Multi-Backpropagation Network In Medical Diagnosis

Wan Hussain Wan Ishak, Fadzilah Siraj, Abu Talib Othman

School of Information Technology,
Universiti Utara Malaysia, 06010 Sintok, Kedah, MALAYSIA
Email: {hussain; fad173; abutalib}@uum.edu.my

Abstract

Backpropagation (or backprop) algorithm is one of the well-known algorithms in neural networks. It is capable to deal with various types of data and also able to model a complex decision system. Some problem domains involve a large amount of data. The bigger the number of input or hidden units is, the more complex the model would be. Hence, reducing the network complexity would be an advantage to the network. This paper proposes a multi-backpropagtion network that reduces the size of a large backpropagation network. The network is divided into several smaller networks, which act as a specialized network. As a result, the raining time would be reduced.

Keywords: *Neural Network, Multi Network, Backpropagation,, Medical Diagnosis*

1.0 Introduction

Artificial Neural Networks or in short, Neural networks (ANNs or NNs) is a computational paradigm that comprises mathematical, statistical, biological sciences and philosophy. These disciplines formulate a formula to form a brain like function, called artificial neuron. Artificial neuron comprises of large number of computational processing elements called units, nodes or cells. Neuron are connected to each other with an associated weight. The weight represents information (or knowledge) being used by the network to solve a problem. Analogously, these processing elements or NN, mimic the processing elements of biological neuron.

NN is defined as a simple processing element whose functionality is based on the human or animal neuron. Simple neuron was introduced by McCulloch and Pitts in 1940s. It consists of input layer, activation function, and output layer. The input layer receives input signal from external environment (or other neuron). Activation function is the neuron internal states that calculates and sum the input signals. The signals are then transmitted to output layer.

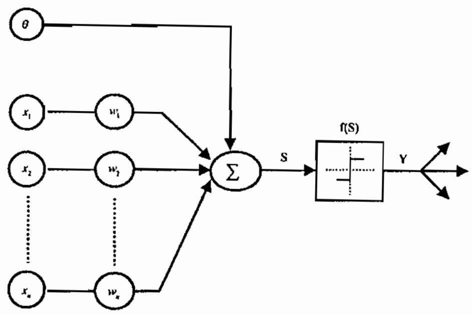

Figure 1: Model Neuron McCulloch-Pitts

NN and its ability to learn is AI technique designed for prediction, clustering and classification tasks. Sarle (1994) describe the usage of NN in three main ways, typically, as models of biological nervous systems and "intelligence", as real-time adaptive signal processors or controllers implemented in hardware and as methods for data analysis. Passold *et al.* (1996) summarized the benefits of neural networks as follows:

- Ability to process a massive of input data
- Simulation of diffuse medical reasoning
- Higher performances when compared with statistical approaches
- Self-organizing ability-learning capability
- Easy knowledge base updating

Training NN could be time consuming and some network could be difficult to train. This circumstance is due to the complexity of the network. Therefore, this paper proposed a multi-backpropagation network to reduce the size of a large backpropagation network. The modification is on the approach of representing the network where the large network is composed into several smaller networks, which act as a specialized network.

2.0 Backpropagation Network

Backpropagation algorithm has been popularized by Rumelhart, Hinton, and Williams in 1980s as a euphemism for generalized delta rule. Backpropagation of errors or generalized delta rule is a decent method to minimize the total squared error of the output computed by the net (Fausett, 1994). The introduction of backprop algorithm (Figure 2) has overcome the drawback of previous NN algorithm in 1970s where single layer perceptron fail to solve a simple XOR problem.

According to Fausett (1994), the aim of backpropagation algorithm is to train the net to achieve a balance between the ability to respond correctly to the input patterns that are used for training (memorization) and the ability to give reasonable (good) responses to input that is similar, but not identical, to that used in training (generalization).

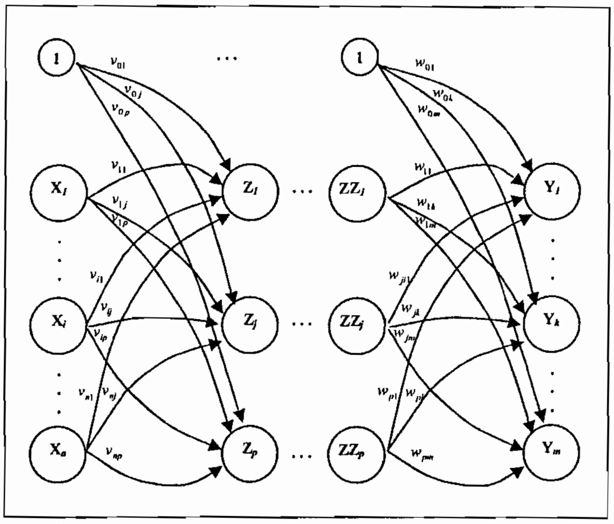

Figure 2: Multi Layer Backpropagation Neural Network

Training Backpropagation network can be divided into three stages (Fausett, 1994):

- Feedforward - the feedforward of the input training pattern
- Backpropagation of error - the calculation and Backpropagation of associated error
- Weight update - the adjustment of the weights

In feedforward phase, each input (X_i) receives an input signal and broadcasts this signal to the hidden units $Z_1 \ldots Z_p$. Each hidden units (Z_p) computes its activation and sends its signal (z_j) to each output unit ($Y_1 \ldots Y_k$). Each output unit (Y_k) computes its activation (y_k) to form the response of the net for the given pattern.

Associated error for the pattern is determined from a comparison between output unit (y_k) and its associate target value t_k. Based on the error, the factor δ_k (k = 1, …, m) is computed. During the backpropagation phase of learning, signals are sent in the reverse direction. δ_k is used to distribute the error from output unit y_k back to all units in the previous layer (hidden units that are connected to Y_k). The error information is then used to update the weights between the output and the hidden layer. In a similar manner, the factor δ_j (j = 1, …, p) propagate the error back to the input layer and update the weights between hidden and input layer. Taken as a whole, the adjustment to the weight w_{jk} is based on the factor δ_k and the activation z_j of the hidden unit Z_j. The adjustment to the weight v_{ij} is based on the factor δ_j and the activation x_i of the input unit.

The training procedure for the backprop is as follows:

Step 0:
Initialize Weight

Step 1:
While stopping condition is false, do steps 2 - 9

Step 2:
For each training pair, do steps 3 - 8

(Feedforward)

Step 3:
Broadcasts input signal (x_i where $i = 1, \ldots, n$) to all units in hidden layer.

Step 4:
Sum weighted input signals for each hidden unit (Z_j, $j = 1, \ldots, p$).

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \,,$$

compute output signal

$$z_j = f(z\_in_j)$$

Step 5:
Sum weighted input signals for each output unit (Y_k, $k = 1, \ldots, m$).

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{jk} \,,$$

compute output signal

$$y_k = f(y\_in_k)$$

(Backpropagation of error)

Step 6:
Compute error information term for each output unit (Y_k, $k = 1, \ldots, m$)

$$\delta_k = (t_k - y_k) f'(y\_in_k)$$

calculate weight correction term ($\mu = [0,1]$)

$$\Delta w_{jk}(t+1) = \alpha \delta_k z_j + \mu \Delta w_{jk}(t)$$

calculate bias correction term

$$\Delta w_{0k} = \alpha \delta_k$$

Step 7:
Sums delta inputs for each hidden unit (Z_j, $j = 1, \ldots, p$)

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk}$$

calculates error information term

$$\delta_j = \delta\_in_j f'(z\_in_j)$$

calculate weight correction term

$$\Delta v_{ij}(t+1) = \alpha \delta_j x_i + \mu \Delta v_{ij}(t)$$

calculate bias correction term

$$\Delta v_{0j} = \alpha \delta_j$$

(Update weights and biases)

Step 8:

Update bias and weights (j = 0,…,p) for each output unit (Y_k, k = 1, …, m)

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$$

Update bias and weights (i = 0,…,n) for each output unit (Z_j, j = 1, …, p)

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$$

Step 9:

Test stopping condition

NNs have been shown as effective implementation in many medical applications such as basic sciences (Prank *et al.*, 1998; Abidi and Goh, 1998), clinical medicine (Bottaci and Drew, 1997; Pofahl *et al.* 1998), signal processing and interpretation (Janet, 1997; Waltrous and Towell, 1995; Lagerholm *et al.*, 2000; Dybowski, 2000) and image processing (Poli and Valli, 1995; Ahmed and Farag, 1998). In addition, several of related research in this applications domain have been discussed in Wan Ishak *et al.* (2001).

3.0 Multi Network

Backpropagation network is capable to deal with various types of data and is able to model a complex decision system. Backpropagation network with hidden layer (or so called multi layer network) is able to process and model more complex problem. However, some problem domains might involve a large amount of data. Backpropagation network with four input units and two hidden units for example required several epochs, which create a complex model. More input units or hidden units could increase the complexity of the model and increase its computational complexity. In other words, an addition to the input unit or hidden unit could increase the model complexity and increase training time. This is because a larger network is more difficult to train.

In Figure 3, we illustrate the problem of multiple logical operation **AND, OR** and **XOR** that is (*A* **AND** *B*) **AND** (*C* **OR** *D*) **OR** (*E* **XOR** *F*). In NN this problem is presented as a set of inputs into the network. The relation between each input is considered understood by the network. As we have six inputs, the total combination would be 64. Hence, training the network to learn all 64 data sets is time consuming where for each epoch the nets have to learn 64 different patterns. Each pattern is feed into the network one at a time and calculates its error information term.

Basically this problem combine several operations and some of the operations is repeated. For example, (*A* **AND** *B*) and (*A* **AND** *B*) **AND** (*C* **OR** *D*) are an **AND** problem. Solving both problems required a logical sets of **AND** (as in Table 3.6(a)). Manually (based on the logical tables) this problem is solved step by step as in figure 3.6. The output from (*A* **AND** *B*) and (*C* **OR** *D*) is **AND**'ed together to obtain its output. Thereafter, its output is **OR**'ed with the output of (*E* **XOR** *F*).

Hence (*A* **AND** *B*) **AND** (*C* **OR** *D*) **OR** (*E* **XOR** *F*) can be divided into three logical networks that are **AND, OR** and **XOR** networks. These three networks will produce a knowledge of **AND, OR** and **XOR** logical operation (which also represent its logical table). Each network will have four sets of data that are [1,1,*t*], [1,0,*t*], [0,1,*t*], [0,0,*t*] where *t* is the targeted value. Training these networks required only several epochs. All three networks will be trained one by one and their weight or knowledge will be stored as the representation of logical operation. Knowledge from **ANDs** network for example can be used for both (*A* **AND** *B*) and (*A* **AND** *B*) **AND** (*C* **OR** *D*) operation.

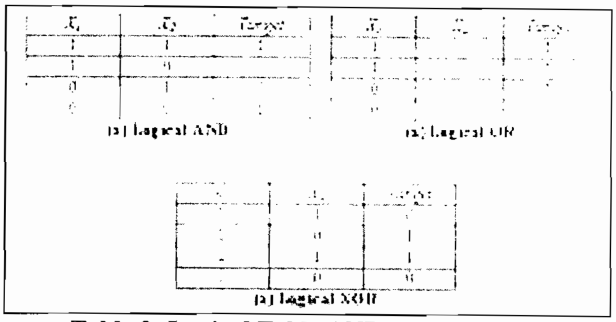

Table 3: Logical Table AND, OR and XOR

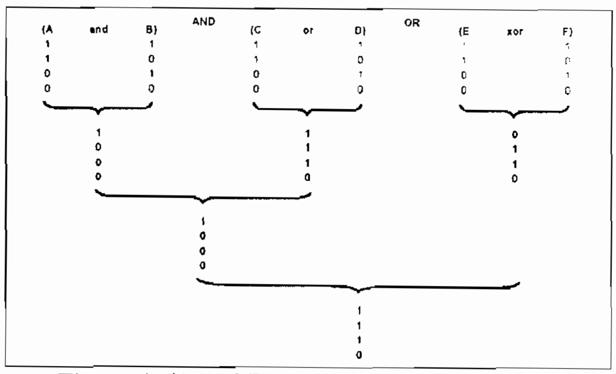

Figure 4: (A and B) and (C or D) or (E xor F) problem

This study aimed to minimize the complexity of data used to train the network. Minimizing the complexity means reducing the complexity of each pattern by normalizing its attributes. Normalization refer here is the same as applied in relational databases where attributes are grouped into several categories to minimize the relationship between attributes. This technique could reduce the redundancy of data. Basically, the idea of multi backpropagation network is similar to the concept of bottom-up hierarchical neural network (see for example Ohno-Machado, 1996). Several specialized networks were constructed to represent certain component of the problem and another network integrates the outputs to produce the final result (Figure 5).

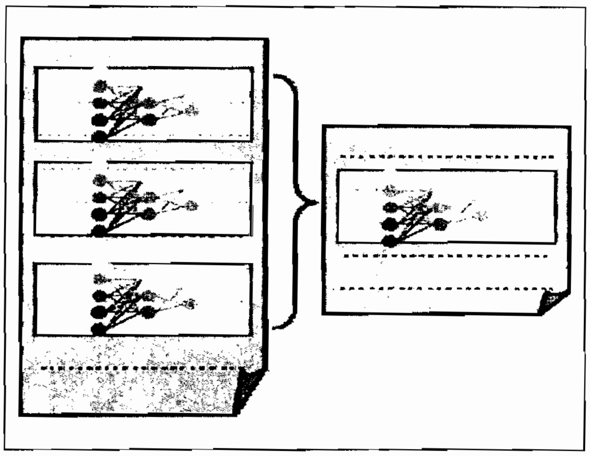

Figure 5: Framework for Multi-Backpropagation representation

In medical, this approach could be implemented to reduce the complexity of patient-disease data. In some cases, diagnosing patient condition involved large amount of data and some of the data may not available at the time of the diagnosis. In current practice this could cause difficulty to the network, as the missing data is treat as missing value.

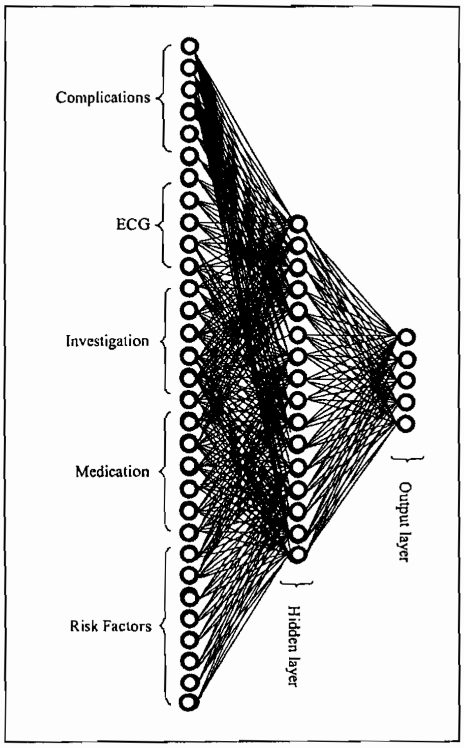

Figure 6: Predicting the presence of Myocardial Infarction

In diagnoses of Myocardial Infraction disease for example (figure 6), many variables have to be considered. However, some of the data are not available as certain investigation or procedure is pending. Therefore, the variables are divided into six different groups that are **COMPLICATIONS, ECG, INVESTIGATION, MEDICATION** and **RISK FACTORS** group (figure 7). Each group will produce an output that represents the group. For example, from the number of the risk factors or its combination, medical practitioner could easily classify patient's risk status whether that patient is high-risk of having MI or not. This interpretation resulted with 0 or 1 based on the interpretation of the condition given. These categories are then grouped into one network to produce its final result (output) (see Figure 7(f)).

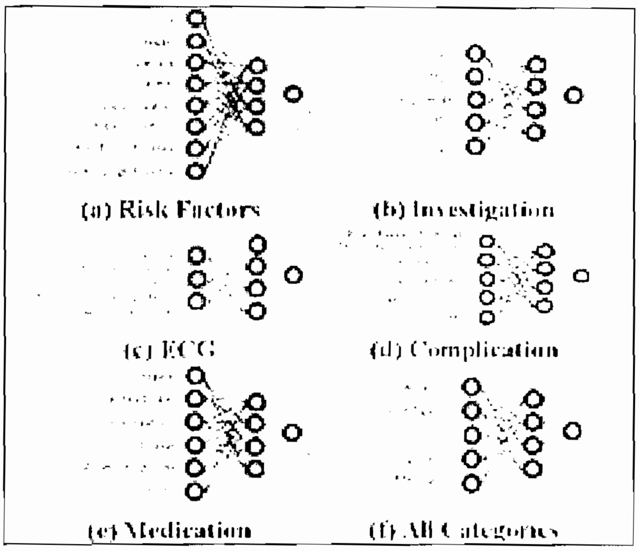

Figure 7: Multi Networks by Category

Compare to the smaller networks, original network (Figure 6) have a large connection link between input, hidden and output layer. The network needs to be train for more epochs so that the network can learn all the patterns. In other words that larger network required more epochs to learn.

4.0 Conclusion

Neural network is one of the promising techniques in many area or applications. In this paper, we have presented backpropagation network with backpropagation learning algorithm. The complete step-by-step procedure for the learning rule is also discussed.

To minimize the complexity of patient-disease data, we proposed the multi network approach. This approach does not require any alterations of the algorithm. The large network is divided into several specialized networks, each network represents one category and finally the output will be integrated to produce the final results. This approach is expected to increase the ability of the network to learn and generalize by giving as much help as it can to the network especially during the training.

5.0 Acknowledgement

Authors would like to thank Ministry of Health for allowing the access and use of Myocardial Infraction data from General Hospital Alor Setar, Kedah.

References

Abidi, S. S. R., and Goh, A. (1998). Neural Network Based Forecasting of Bacteria-Antibiotic Interactions for Infectious Disease Control. In 9th World Congress on Medical Informatics (MedInfo'98), Seoul August 18-22.

Ahmed, M. N., and Farag, A. A. (1998). Two-stage Neural Network for Medical Volume Segmentation. Accepted for Publication in the Journal of Pattern Recognition Letters, 1998.

Bottaci, L., and Drew, P. J. (1997). Artificial Neural Networks Applied to Outcome Prediction for Colorectal Cancer Patients in Separate Institutions. Lancet, Vol. 350, Issue 9076, pp. 469-473.

Dybowski, R. (2000). Neural Computation in Medicine: Perspective and Prospects. In Malmgren, H., Borga, M., Niklasson, L. (eds.) Proceedings of the ANNIMAB-1 Conference (Artificial Neural Networks in Medicine and Biology), Goteborg, 13-16 May 2000. Springer, pp. 26-36.

Fausett, L. (1994). Fundamentals of Neural Network: Architectures, Algorithms and Applications. Prentice Hall; Englewood Cliffs.

Janet, F. (1997). Artificial Neural Networks Improve Diagnosis of Acute Myocardial Infarction. Lancet, Vol. 350, Issue 9082, pp. 935.

Lagerholm, M., Peterson, C., Braccini, G., Edenbrandt, L., and Sörnmo, L. (2000). Clustering ECG Complexes Using Hermite Functions and Self-Organizing Maps. IEEE Transactions on Biomedical Engineering 47, 838-848

Ohno-Machado, L. (1996). Medical Applications of Artificial Neural Networks: Connectionist Model of Survival. Ph.D Dissertation. Stanford University.

Passold, F., Ojeda, R. G., and Mur, J. (1996). Hybrid Expert System in Anesthesiology for Critical Patients. In Proceedings of the 8th IEEE Mediterranean Electrotecnical Conference - MELECON'96 (ITALIA), Vol. III, pp. 1486-1489.

Passold, F., Ojeda, R. G., and Mur, J. (1996). Hybrid Expert System in Anesthesiology for Critical Patients. In Proceedings of the 8th IEEE Mediterranean Electrotecnical Conference - MELECON'96 (ITALIA), Vol. III, pp. 1486-1489.

Pofahl, W. E., Walczak, S. M., Rhone, E., and Izenberg, S. D. (1998). Use of an Artificial Neural Network to Predict Length of Stay in Acute Pancreatitis. Presented as a poster at the 66th Annual Scientific Meeting and Postgraduate Course Program January 31-February 4.

Poli, R, and Valli, G. (1995). Optimum Segmentation of Medical Images with Hopfield Neural Networks. Technical Report CSRP-95-12. The University of Birmingham.

Prank, K., Jurgens, C., Muhlen, A., and Brabant, G. (1998). Predictive Neural Networks for Learning the Time Course of Blood Glucose levels from the Complex Interaction of Counterregulatory Hormones. Neural Computation, Vol. 10, Issue 4, pp. 941-954.

Sarle, W. S. (1994). Neural Networks and Statistical Models. Proceedings of the Nineteenth Annual SAS Users Group International Conference, April, 1994.

Silipo, R., and Marchesi, C. (1998). Artificial Neural Networks for automatic ECG analysis, IEEE Transactions on Signal Processing, Vol. 46, n. 5, pp. 1417-1425.

Waltrous, R., and Towell, G. (1995). A Patient-Adaptive Neural Network ECG Patient Monitoring Algorithm. Computer in Cardiology, Vienna, Austria.

Wan Ishak, W. H., Siraj, F., Othman, A. T. (2001). Neural Network in Medical Application: A Review. Simposium Kedua Penyelidikan Pengurusan Siswazah (16-17 Oct. 2001). Universiti Utara Malaysia